



# Pyramid Security Hardening Guide

Options and approaches to enhance the security of your Pyramid installation.

Version 1.0

© Pyramid Analytics 2022-2023



## Contents

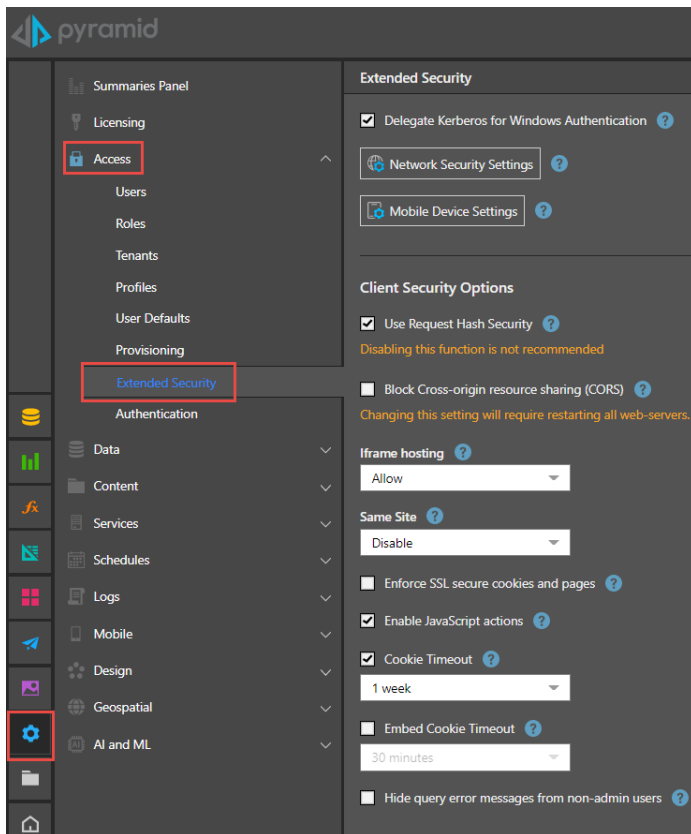
Overview .....	3
Settings within the Application.....	3
Network Security .....	4
Mobile Security .....	5
Client Security .....	6
Multi-Factor Authentication (MFA) .....	7
How MFA Works .....	7
Why is MFA more secure .....	<b>Error! Bookmark not defined.</b>
MFA vs Mobile Device ID Check.....	7
Enabling MFA .....	7
TOTP Authenticator Apps and Enrollment.....	7
Resetting MFA tokens for Admin Users .....	8
Hardening the Hosting Environment .....	9
Hardening the Web Server.....	9
Microsoft Windows IIS.....	9
NGINX Web Server .....	9
Hardening the Database Repository .....	9
Hardening the hosting servers .....	9
Securing Kubernetes .....	9
Summary .....	10

# Overview

Pyramid incorporates and supports numerous security elements to ensure a safe, secure environment with robust performance. This document provides details on the various “hardening” security features available in Pyramid as well as recommendations for securing the environment(s) that Pyramid may operate in.

## Settings within the Application

Numerous security options are available within the application itself. Administrators can access these options through the “Extended Security” features from the Access menu via the Admin Console (red boxes).

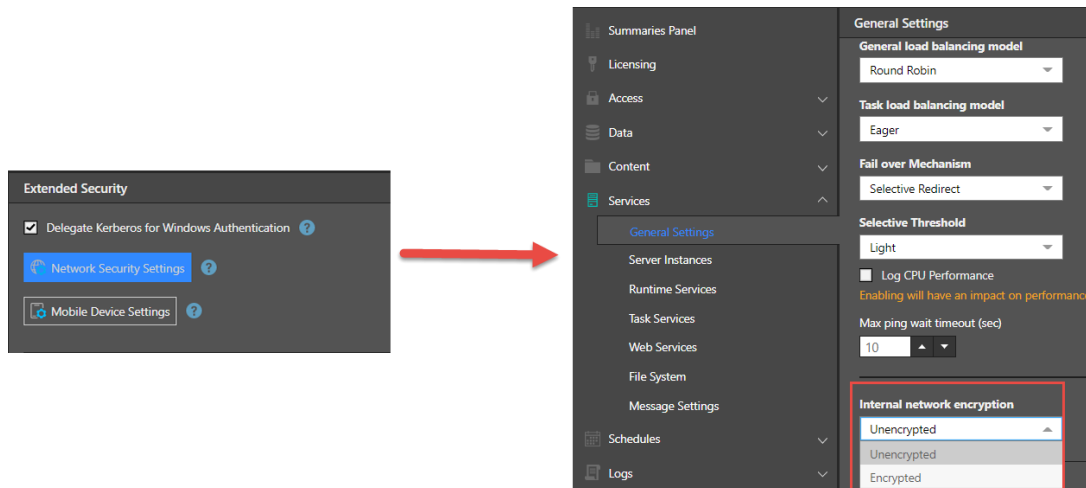


## Network Security

Security protects the usability and integrity of a company's infrastructure by preventing the entry of a wide variety of potential threats. Network security is a set of configurations designed to protect the confidentiality and accessibility of the application.

Pyramid provides an option to encrypt the internal network to further strengthen internal security.

Clicking on the Network Security Settings button will take you to the General Settings within the Services menu, where you can change the internal network encryption (red box).

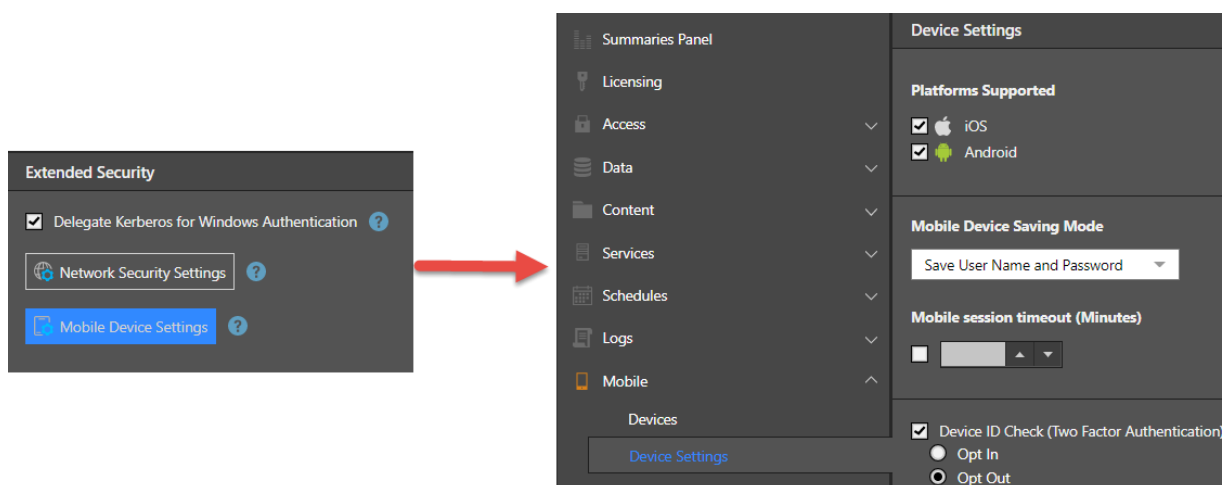


- Encrypted:** When selecting this option all internal communication between the pyramid services will be encrypted. Enabling this option can cause a degradation in the performance due to the encryption layer. This option is recommended only if internal communication security between the services is required, and Pyramid is not installed behind a secure network.
- Unencrypted:** This is the default option when no additional security is required.

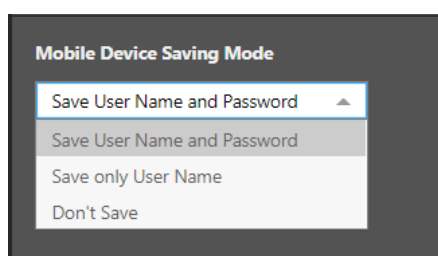
## Mobile Security

Mobile users are an additional risk due to the easy accessibility of mobile devices. Pyramid provides extended security for mobile devices.

Clicking on the Mobile Device Settings button will take you to the Device Settings within the Mobile menu, where you can change the platforms supported, saving mode, session timeout and the Device ID Check.



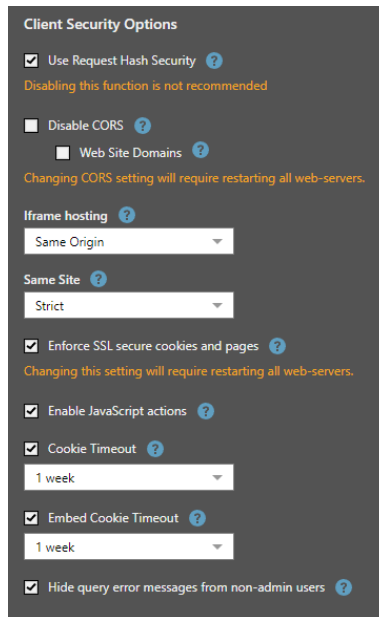
- **Platforms Supported:** Allows admin to limit device access for either iOS and Android.
- **Mobile Device Saving Mode:** Allows admin to restrict the saving of user and password on the mobile device. Admins can enable users to save username and password, user name only, or disable the ability to save user name or password. Disabling the user's ability to save their username and password increases security, but will force users to re-enter their user name and password when logging in to the system.



- **Mobile session timeout:** Allows admins to check the timeout box to specify the amount of time before the mobile session is timed out. A shorter timeout period will safeguard the device forcing users to re-login more often user having to login again.
- **Device ID Check:** selecting the device ID check turns on MFA capabilities from the mobile by requiring two sets of details to be sent from the mobile app upon login: the user's credentials and the mobile device's ID. When selecting the device ID check the administrator can choose between one of the two choices
  - **Opt In:** devices which are added to the system are DISABLED by default until an admin has enabled the device on the framework.
  - **Opt Out:** devices which are added to the system are ENABLED on the system by default and admin can disable specific device or devices .

## Client Security

In addition to network and mobile device security, Pyramid provides many additional client security options, for securing the way the web-based clients operate.



- **Use Request Hash Security:** Option to add a hash check to key client-side functions to ensure that only authorized users are performing authorized activities on relevant content.
- **Disable CORS (Cross-origin resource sharing):** Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served. This capability is needed in Pyramid specifically when using the embedding capabilities. An admin can choose to Disable Cross-origin resource sharing (CORS) and prevent Pyramid from accepting requests from other domains. Note that if this option is enabled, embedding capabilities will be disabled.
  - **Web Site Domains:** If CORS is left enabled, then a white list of web domains should be provided that can be used for cross domain access, to prevent any degradation in client security.
- **Iframe hosting:** Option to disable IFrame hosting.
  - Allow: enables IFrame hosting. They are commonly used for advertisements, embedded videos, web analytics and interactive content.
  - Deny: This blocks all IFrame hosting. If IFrame hosting is blocked, IFrame embedding capabilities will be disabled.
  - Same Origin: enables IFrame hosted in the same website domain as Pyramid only
- **Same Site:** Prevents the browser from sending a cookie with cross-site requests. This mitigates the risk of cross-origin information leakage and cross-site request forgery attacks.
  - Disable: allows cookies to be sent
  - Lax: the cookie is sent with GET requests or top-level navigation with a safe HTTP method.
  - Strict: stops the cookie being sent by the browser to the target site in all cross-site browsing contexts, including when following a regular link.
- **Enforce SSL secure cookies and pages:** Ensures the application can only operate with SSL encrypted websites only (HTTPS), the application will be blocked from operating with plain HTTP
- **Enable JavaScript actions:** Enabling this option allows users to configure JavaScript actions as defined in Discover or Present, to execute a script in the browser. This could provide a security risk. This option must be enabled to configure JavaScript actions that have been defined in Discover or Present to execute a script in the browser.
- **Cookie Timeout:** Options to enforce cookie expiration with the ability to set the cookie timeout period. This ensures users must login to the application again when a cookie is marked as expired. The timeout period can be set to a period from 30 minutes to 12 months.
- **Embed Cookie Timeout:** Options to enforce the embedded token to expire - only relevant if using embedded content. In this scenario, you can use the `pyramid.authFailure` API; you can implement the behavior of this function. For example, you may want to redirect users to the Pyramid login page or show them a message. This ensures users must login to the application again when an embedded token is marked as expired. The timeout period can be set to a period from 30 minutes to 12 months.
- **Hide Query Errors:** This is a precautionary switch to hide any query related errors, and associated query details, from non-admin users.

# Multi-Factor Authentication (MFA)

MFA ensures that even if a user's username and password are stolen, access to Pyramid is blocked without the independent security code from the authenticator app. This code changes regularly, so it has a short lifespan before it is useless. Admins can also reset the MFA token associated with a given user, further ensuring that if the authenticator app is compromised, access can be centrally blocked.

MFA is highly recommended to secure access to any web-based solution.

Pyramid offers Multi-Factor Authentication as an out-of-the-box feature when using the internal authentication provider (via the "FORMS" option), LDAP or Active Directory when used with forms (not Basic or Windows Auth). When using external authentication providers (like Azure AD or OKTA), the MFA mechanism is provided by those solutions and does not utilize Pyramid's internal MFA engine.

## How MFA Works

Pyramid's multi-factor authentication uses Time-based One Time Passwords ("TOTP") - which prompts an enrolled user to put in a machine generated key to login to Pyramid. This unique key is generated by special applications (usually on the user's Smart Phone) and changes every 30 seconds. The user is then asked to login with their standard credentials (username and password), together with the 6-digit TOTP code.

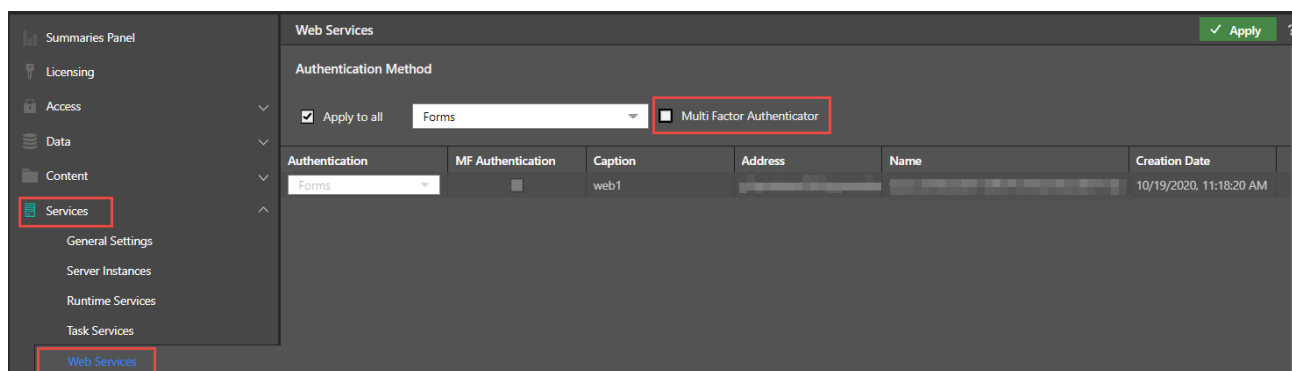
MFA ensures that even if a user's username and password are stolen, access to Pyramid is blocked without the independent security code from the authenticator app. This code changes regularly, so it has a short lifespan before it is useless. Admins can also reset the MFA token associated with a given user, further ensuring that if the authenticator app is compromised, access can be centrally blocked.

## MFA vs Mobile Device ID Check

The MFA capability in Pyramid is independent of the device ID check performed on all mobile connections to Pyramid. Both deliver a two-factor authentication model ensuring that the user needs to know and have something to log into the application.

## Enabling MFA

Admins enable MFA in the web services panel from the admin console.



Here, they can turn it on for the entire system if forms is being used on all web servers, or for each individual web server (as needed). Importantly, this item will have no effect if users authenticate and access the system via an SSO authentication framework (SAML, OpenID etc.).

## TOTP Authenticator Apps and Enrollment

Users need to first download an authenticator app - typically to their smart phone. Both Microsoft and Google have free 'authenticator' apps for both Android and iOS.

Once MFA is enabled, the first time a user logs into Pyramid, they will be asked to enroll into the MFA engine by scanning a QR code into their authenticator app. The app will immediately add a new key into the app and respond with a 6-digit code. Users will then be prompted to supply this 6-digit code every time they log into Pyramid.

Key things to note:

- The code changes every 30 seconds.
- The TOTP system is based on time. So admins need to ensure that the time stamp on servers is perfectly synchronized with world time.

- If a user loses their authenticator access, their MFA token can be reset in the user admin tools. This will force the user to re-enroll the next time they login.
- The TOTP prompt will not appear unless the user's access token into Pyramid has expired or does not exist. Admins can ensure tokens expire regularly by setting a cookie timeout in extended security settings.

## Resetting MFA tokens for Admin Users

The System Maintenance Tools are available for admin users as a fail-safe in case their own MFA tokens are corrupt or lost.



# Hardening the Hosting Environment

The following sections cover techniques and ideas for hardening the environment that hosts a Pyramid instance. As such, it refers to applications and technologies outside of Pyramid itself.

## Hardening the Web Server

The web server is the engine that powers the HTML5 client interface in your browser. Pyramid embeds its own internal web server engine for this operation. However, SSL encryption and decryption cannot be implemented on Pyramid's internal web server engine. Other web server "options" can be deployed to facilitate hardening of the web server using SSL encryption and decryption.

### Microsoft Windows IIS

Microsoft IIS is an option in Windows deployments. It acts as a 'reverse proxy' server and will allow for the easy use of external host header URLs and SSL certificates and HTTPS binding on the relevant ports (like 443). For IIS to operate, the URL Writer and ARR modules for IIS need to be installed. Once installed the proxy server settings need to be enabled. When choosing IIS as the "Web Server" - these steps are automated for you during the installation.

- [Click here](#) for more details on setting up an IIS based website.

### NGINX Web Server

NGINX acts as a 'reverse proxy' server, allowing external host header URLs and SSL certificates and HTTPS binding. It is a recommended option for Linux deployments.

- [Click here](#) for more details on setting up a NGINX based website.

## Hardening the Database Repository

The database repository houses all the content and settings in the system. To secure the database repository, the server housing the database repository should be secured by a firewall. For customers using existing SQL server and Oracle databases for the repository, the security settings applied to the database will automatically apply to the database repository.

- For users using Postgres for their database repository, Postgres is installed during the installation. The Postgres connection should be [secured with SSL](#).
- To prevent access from undesired external users, IP addresses should be [restricted from the Postgres connection](#).
- Access to the server and database should be secured via users and passwords.

## Hardening the hosting servers

To provide improved security for the hosting servers the following recommendations should be followed:

- A vigorous password strategy should be applied to ensure strong passwords are enforced and passwords are changed on a regular basis.
- Regular Scanning and Testing of the environment should be performed.
- Firewall Protection should be applied.
- Update Software Regularly to ensure latest security and other patches are applied.
- No direct access to login to the servers from outside the network – enforce use of VPN.

## Securing Kubernetes

Kubernetes security is mostly managed by third-party apps and tools. The description of how to employ these tools and techniques is beyond the scope of this document. For more details on hardening Kubernetes, you can follow the recommendations [suggested here](#).

## Summary

Pyramid's security hardening features provides administrators the ability to lock down their environment. Pyramid's network security feature provides an option to encrypt the internal network. Mobile Device Settings allow administrators to change the platforms supported, saving mode, session timeout and the Device ID Check.

The additional client security options include the ability to allow users to add a hash check to client-side functions; blocking of cross origin resource sharing; and the ability to disable IFrame hosting. Cookie management prevents the browser from sending a cookie with cross-site requests; can ensure the application can only operate with SSL encrypted websites if required; and, provides options to enforce cookie expiration.

Pyramid also offers Multi-Factor Authentication as an out-of-the-box feature where relevant and appropriate.

There are numerous techniques for hardening the hosting environment for Pyramid. Most of these are beyond the scope of this document, however, administrators should take care to lock down the web servers, the repository database, and the machines hosting the main Pyramid services, including Kubernetes.